HESDC 1997

MODELING THE IMPACT OF PACKAGING DURING HIGH-LEVEL SYSTEM DESIGN: The Integration of Physical Partitioning Into Virtual Prototyping

Paul E. Stoaks and Peter A. Sandborn
Savantage, Inc.
3925 W. Braker Lane, Suite 325
Austin, Texas 78759-5321 USA
(512) 305-0050
{pstoaks, sandborn}@savantage.com

ABSTRACT

As system designers strive to meet the conflicting objectives of cheaper, smaller, and faster electronic systems, the impact of packaging decisions on the design outcome is growing. As higher density systems become prevalent, virtual prototyping activities (functional decomposition, logical partitioning, hardware/software partitioning) must be integrated with physical implementation details at an earlier design stage to insure design success. "Physical partitioning" involves determining what components are assembled into which ICs, modules, or boards, and what type of implementation technologies are used to meet design for cost, and design for manufacturability goals. This paper discusses the role of "physical partitioning" in system virtual prototyping and new software developments that integrate physical partitioning with other virtual prototyping activities.

INTRODUCTION

Electronic systems are composed of a hierarchy of chip and packaging technologies including bare die, die bonding, single chip packages, modules/boards, connectors, backplanes, and cooling structures. One of the keys to successful electronic system design is to find the optimum combination of components and technologies in this hierarchy to meet the cost and performance goals set for the system.

The need to simultaneously improve the performance of designs while reducing their size and cost makes system design difficult. In particular, the drive to larger die, faster clock rates, and higher system densities push designers to use advanced packaging and interconnect concepts such as multichip modules (MCMs), advanced single chip packages (ball grid arrays, chip scale packages), and micro-via boards. If it were not for the economic constraint (the need to make the product cheaper), the choice would be easy; Advanced packaging and board technology would be used throughout designs challenged with high performance and limited space requirements. For example, MCMs are the best choice for high performance and high density systems, and they accommodate very large IC's nicely. Unfortunately, the dominant system driver is usually cost, and MCMs are still relatively expensive.

Therefore, today's system designer is faced with the fundamental tradeoff of advanced packaging vs. system cost. Since today's systems are extremely large and complex, choosing one implementation technology for the whole system is typically not the best answer. Packaging the entire system using the highest performance technologies results in wasted cost since low performance sub-systems are implemented with expensive packaging technology. System designers today must partition the design into physical sub-assemblies, then choose the right technology for each sub-assembly to optimize the size and performance characteristics against total system cost. This is the essence of the physical partitioning problem.

SYSTEM VIRTUAL PROTOTYPING

To deal with the fundamental system cost vs. advanced packaging tradeoff, a design flow like the one shown in Figure 1 may be used. Figure 1 shows a system design flow starting with product requirement capture and ending with the delivery of the completed electronic system. The significant difference between this design flow and a typical flow used by system designers today is the addition of the system virtual prototyping activity and the consideration of physical partitioning concurrent with system logical partitioning and hardware/software partitioning.

Definitions of "virtual prototyping" abound. They tend to focus on "virtual reality", "simulation", etc. The definitions agree that a virtual prototype is built on a computer, is built before a physical prototype would/could be built, and allows for evaluation of certain aspects of the design that in the past would only have been possible with a physical prototype. If the virtual prototype includes behavior, that behavior is consistent with the actual behavior of the design in the chosen environment. In short, a virtual prototype is a computer model of a system that is constructed prior to the design and implementation of the system and that is presented in such a way that it can be evaluated by humans and by algorithmic methods. The earlier in the design cycle that this prototype can be constructed and evaluated, the more value it brings to the design process.

There are four major activities involved in creating a virtual prototype. Functional decomposition is breaking the system down into its functional components and defining their behavior. Hardware/Software partitioning is choosing which functional components are met with software and which with hardware. Logical partitioning is determining the system's logical architecture, or how each function will be realized using hierarchical electronic building blocks. Physical partitioning is the assignment of logical blocks to physical assemblies and the process of selecting the implementation technologies for those assemblies. The development of a virtual prototype may combine top-down and bottom-up (especially when re-use is involved) design strategies. With a top-down strategy, the system is described functionally, logically, and physically at a high level of abstraction and successively refined as development progresses. Evaluation of the virtual prototype against the system requirements can be performed at each step along the way, insuring that the system design stays on course.

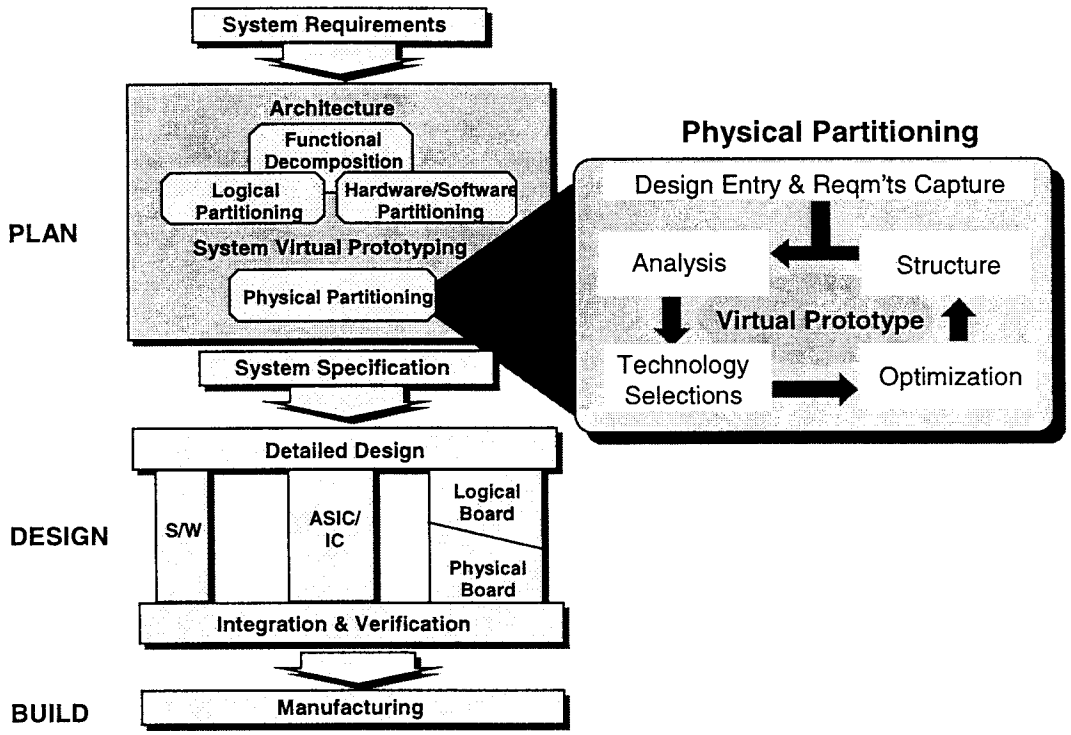

Figure 1: System design flow using a system virtual prototyping methodology

PHYSICAL PARTITIONING

Most design methodologies today fall short in the physical partitioning activity. Thorough analysis of the tradeoffs between a variety of implementation strategies is difficult and seldom performed, resulting in *ad hoc* decisions which do not produce optimized systems. The lack of a rigorous physical partitioning activity is due both to the lack of tools for performing these physical tradeoffs and to the lack of accurate manufacturing data to drive the tradeoffs. Poor tradeoff analysis often creates a real problem in design because up to 80% of the system cost and performance becomes locked-in prior to the start of detailed system design. Attempting to make packaging and technology decisions later in the design process often results in time consuming and costly redesign.

When there were fewer packaging and technology options available, rigorous analysis was less critical. Each design was similar to the previous one, and physical partitioning strategies were fairly obvious. Unfortunately with today's technology choices, it is not obvious what physical architecture and mix of technologies optimizes the cost of the system. Additionally, as the complexity of systems has grown, optimizing a system can no longer be accomplished by optimizing its parts (divide and conquer), rather, the entire system must be considered during optimization. In a complex size, cost, and performance driven system, the partitioning of functionality between ICs, and printed wiring boards (PWBs), and the selection of technologies for each partition (chip-on-board, microvia boards, etc.) are critical to meet the design goals. A good design requires multi-dimensional tradeoff analysis in an environment where partitioning options can be quickly explored before the large investment is made in detailed design and implementation.

Within the IC design discipline, only limited attention has been paid to physical partitioning. During floorplanning, physical partitioning can be evaluated and changed in order to meet design goals, and during synthesis/optimization, physical constraints are considered for logical partitioning. Both of these physical partitioning aids are limited in their ability to influence the design outcome. Floorplanning is limited by the analyses available for evaluation of a design architecture, and by the fact that it depends upon a detailed netlist. This latter dependency pushes evaluation of physical partitioning to a point after the key technology choices have already been made and the floorplanner becomes little more than a pre-layout placement optimizer. Changes to the design's physical hierarchy during floorplanning are possible, but require that the netlist hierarchy be back annotated, which is very difficult and time consuming. A number of PWB/system floorplanners are appearing on the market to bring floorplanning capabilities to the system designer. While this will provide some benefit, the reliance on a netlist and detailed component definitions seriously reduces its impact on the design cycle. In a system design, physical partitioning is such a critical part of the design process that it cannot wait until a detailed netlist is available. Physical partitioning tradeoff analyses must also consider a wider range of system drivers than simply size, delay, and power.

To date synthesis/optimization has been unable to consider more than a few constraints simultaneously (size/delay/power, for instance) and as a result, important design drivers such as test cost and ease of assembly are missed. This makes the methodology difficult to apply to system design. A systems view of partitioning requires the consideration of several issues: 1) modified partitioning objectives (the basis for IC partitioning may not be appropriate for systems containing multiple chips), 2) the treatment of additional technologies such as connectorization and enclosures, and 3) connection topology. Module or board level partitioning is essentially analogous to its IC-

level counterpart, although more interdisciplinary physical constraints must be considered and the basis for partitioning becomes broader. The use of objective functions designed to help partition functionality into chips may be appropriate for traditionally packaged systems (surface mount and through-hole) but is dangerous for high-density packaging systems such as multichip modules. In traditionally packaged systems, the cost of the system tracks the accumulated cost of fabricating the chips, however, in advanced systems where bare die are used or extensive test and rework of modules is required, the system costs often do not track the component fabrication costs. High-density packaging systems that contain bare die must include cost modeling that can assess the impact of Known Good Die (bare die test and burn-in) and the possibility of performing repair and rework operations. If these critical processes are not included then automated partitioning exercises are of little more than academic interest.

An example of a critical element of system-level physical partitioning that has no analog in IC-level partitioning is connection topological relationships. Connection topology is the physical orientation of one board or module to another. There are three topologies which are applicable to inter- module or board connections: plane-in-plane (single chip package or 3D stacking), edge-to-plane (edge connector), and edge-to-edge. (Wilson 1982) The connection topology used has no relationship to the number of connections required, but will determine the number of interconnect crossovers (a crossover is created when one connection between modules or boards crosses over another). (Sandborn 1995) Crossovers tend to add complexity that penalizes the system's size, cost, reliability, and electrical performance. The number of connection crossovers can be minimized by appropriately partitioning the problem. Closely related to topology is disconnection[1]. Disconnection is the methodology and physical realization of how separable physical entities are combined to form a system. Placement of components relative to each other within a physical entity is an important contributor to several performance measures including routability, electrical, thermal, and manufacturability.

Minimally, a physical partitioning system must consider the interrelated components shown in Figure 1:

Structure: Structure represents the physical architecture of the system. During physical partitioning, the functional components of the design are assigned to physical partitions such as ICs, MCMs, few chip packages, PWBs, etc. In addition to partitioning, structure includes placement (the orientation of adjacent components to each other), topology (the orientation of adjacent boards to each other), disconnection (designing subsystems to be disconnectable), and reuse (dividing systems into parts that can be reused in other systems). As the physical partitioning activity progresses, different system structures may be evaluated, necessitating the movement of functional components between physical partitions and changes to component/sub-assembly placement within the partitions.

Technology Selections: Technology characterization is at the core of physical partitioning. The degree that physical partitioning depends on technology decisions far exceeds the technology requirements of the other portions of the virtual prototyping solution. The technologies that must be characterized and encapsulated in libraries to support physical partitioning include components, substrates, materials, connectors, packages (for packaging bare die), and process flows for fabricating and assembling systems. In addition to the data itself, this category includes the management of the data. Technology data management methodologies must allow the physical partitioning activity to access current technology data quickly and accurately.

Analysis: Analysis comprises all the "Design for X" activities (manufacturability, reliability, environment, testability, design-to-cost, etc.) plus all performance estimation activities (electrical, thermal, etc.). The analysis component can be populated with point design tools (simulators) and/or estimation-level advisors. Experience tends to indicate that most of the analyses performed in this area should lead back to cost at some level and must be tightly coupled to manufacturability. This category includes post manufacturing activities: design for recyclability, disassembly, serviceability, maintainability, upgradability, etc, and lifecycle analysis (LCA) methods for systematically assessing material use, energy use, waste, services, processes and technologies over a product's entire life.

Optimization: Optimization is a management framework within which all the partitioning and "Design for X" activities are performed. Optimization does not necessarily imply that the system must choose the optimum design specification without user involvement, but rather, it represents tools to aid the user in collaboratively optimizing their system's physical implementation. Optimization generally includes objective function formulation, allocating and budgeting, prioritization, and the management of constraints and requirements.

The four categories of activities above must seamlessly interact with each other in order to provide an efficient physical partitioning solution.

INTEGRATING PHYSICAL PARTITIONING INTO SYSTEM VIRTUAL PROTOTYPING

Solutions already exist or are emerging for each of the virtual prototyping components shown in Figure 1. Functional design and simulation/verification environments like Nu Thena's Foresight™ suite (Vertal 1997) allow the user to develop the functional and logical architecture of the design. The behavior and performance of the system can be verified by executing the resulting virtual prototype.

In the hardware planning space, a number of different types of tools have attacked parts of the problem, but none have provided a complete solution. All activities in the planning space attempt to manipulate system elements (e.g. function, block, die, chip, module, PWB, etc.) within different views in order to help optimize the system. For example, floorplanning tools relocate components in two-dimensional placement (block within a chip or chips within a board) to change the system's cost and performance. Other tools in the planning space perform more abstract manipulations, such as changing the definition of a bare die in the technology view (e.g. change its

---

[1] Connecting objects together is easy. Connecting objects together under constraints that require the components to be disconnected (for upgradability, supportability, maintainability, testability, etc.) requires a different mind set. Therefore we refer to the activity of connecting components in a disconnectable fashion as the "Disconnection" problem.

bonding technology) or a printed wiring board in the manufacturing view (e.g. use manufacturing facilities in Singapore instead of Denver.) Most tools are limited to manipulating design factors within a single view. Figure 2 shows some of the planning views and which EDA tool types manipulate those views. Component set synthesis tools such as OmniView's Fidelity™ optimize component selection. Architectural planning tools are intended to optimize logical and hardware/software architecture. Packaging tradeoff tools such as Savantage's SavanSys™ (see Appendix) provides facilities for optimizing technology and manufacturing views of the design. System level floorplanner/placement tools like Xynetic's EDAnavigator™, and Cadence's BoardQuest™, optimize 2D placement. A physical partitioning solution for use in the virtual prototyping environment must allow relocating system elements in multiple views concurrently to effectively facilitate system level optimization.

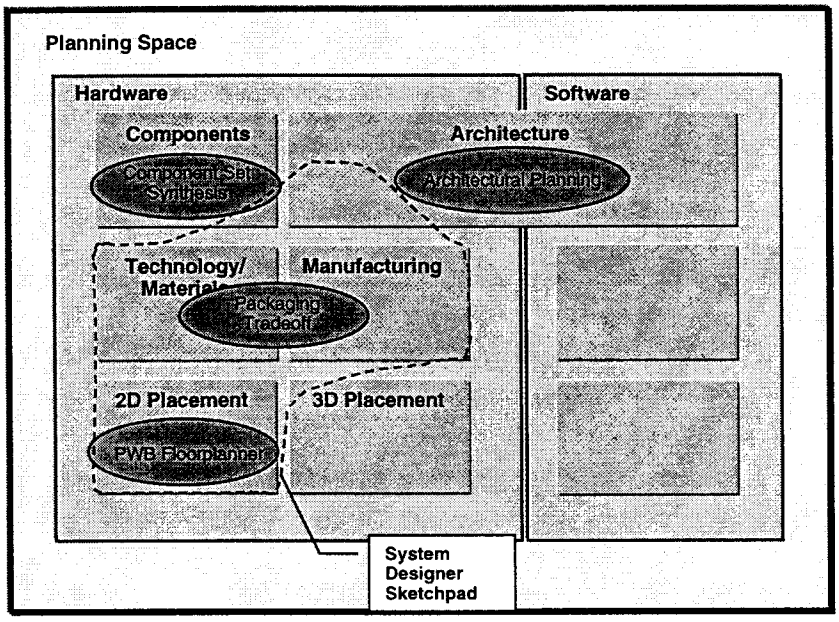

Figure 2: Mapping of EDA tool types to system planning activities. System Designer Sketchpad allows design optimization via concurrent manipulation among multiple views. System Designer Sketchpad is described later in this paper.

Over the last two years, Savantage has been gathering system physical partitioning requirements from customers and industry contacts. This process has included both discussions on the subject and cooperatively performing physical partitioning on real designs. From these exchanges, Savantage has learned that an effective system physical partitioning solution must meet the following requirements:

- Hierarchical Block Diagram Input. Minimally, physical partitioning requires that the system's logical architecture[2] be available as a starting point. Since a number of the tools already available in the virtual prototyping flow provide block diagram input capability and produce HDL or a rough netlist, it would be sufficient to provide interfaces that can produce a hierarchical block diagram from these inputs. A simple, easy to use, block diagram editor in the physical partitioning environment is also necessary for manipulation of imported block diagrams or capture of block diagrams in the absence of one of these other tools. Graphical entry is the preferred capture mechanism for hierarchical system architecture descriptions at this level.

- Physical Structure Definition. In order to be able to accurately model an entire system, the user must be able to define physical partitions that are implemented as ICs, MCMs, few chip modules, PWBs, sub-assemblies, cages, etc. and assemble them into a physical hierarchy. Architectural elements from the block diagram will then need to be assigned to these physical partitions. It must be possible to create one-to-one mappings where a single architectural element (block or part) is assigned to a single physical partition. It must also be possible to create many-to-one mappings where multiple architectural elements are assigned to a single physical partition. A graphical drag and drop interface is the preferred interaction paradigm for this task. Placement and assembly editing functionality is required to allow for positioning the components and sub-assemblies relative to each other. While this placement task need not occur in a 3D environment, the editor must support the placement of objects in 3-space. In order to support re-use, the physical partitioning environment must support the import of completely and partially implemented physical partitions from common design environments.

- Technology Selection/Input. The physical partitioning environment must be able to access a variety of different information repositories in order to obtain part, material, package, design rule, cost, and assembly flow information. This requires that interfaces to common component management systems and EDA environments be provided. In addition, it must be possible for corporate CAD groups to easily build interfaces to provide access to proprietary information management systems. For

[2] The logical architecture of the system consists of the system's logical hierarchy, and gross connectivity (busses, bundles, and critical nets).

information that is not electronically available, the physical partitioning environment must support straightforward entry mechanisms that allow partial to complete definition of technology parameters. Finally, for attributes that are not available in any form, the system must be able to provide reasonable defaults or estimations that may be used until correct values can be obtained. The physical partitioning environment should also allow for the encapsulation of certain "expertise" or "knowledge" in models that can be compiled into libraries for tradeoff analysis. This allows tradeoff analysis in areas where the user may not have direct knowledge, such as manufacturability or assembly location selection.

- Tradeoff Analysis. At a minimum, the following tradeoff analyses must be supported: size, routability, I/O, cost, critical net delay, and manufacturability. With today's designs, thermal and EMI analyses are becoming extremely important as well. The analysis capability should be extensible through 3rd party plug-ins. For the analysis capability to be useful early in the design cycle, it is important that all analysis modules be able to provide useful estimates with incomplete inputs. Each output value should be accompanied by an uncertainty or confidence metric to help the user gauge it's usefulness. As important as the electronic analysis is, it is also important that the partitioning environment provide output that can easily be evaluated by humans. To this end, the physical partitioning environment should provide 2D and 3D visualization of physical partitions and whole assemblies. This allows the user to view the physical virtual prototype and detect potential problems that are not readily apparent by algorithmic means.

- Collaborative Optimization. The physical partitioning environment must support a collaborative optimization paradigm. One of the most difficult problems in optimizing a design is the determination of which of factors to tune in order to achieve a desired outcome. The environment should support sensitivity analysis to aid in the determination of what factors most influence a given metric. In addition, it is important to be able to run consecutive analyses where a parameter is varying over a range.

- Design Flow Output. The physical partitioning solution must be able to communicate results with other design tools in the virtual prototyping environment. Critical net delay and any other parameters determined by the physical architecture and technology selection should be exportable. Many of the results obtained during the physical partitioning activity are contained in the definitions of the physical partitions and the documentation of the decisions made. As a result, the output is primarily a specification for the detailed design activity rather than some set of files that can be read by downstream tools. However, design rule and layer stackup definitions, and potentially component placement information, can be passed to floorplanning and layout tools. It is possible that certain assembly information can also be exported to mechanical CAD systems as a starting point for mechanical design. A broader usage of physical partitioning as a methodology will further flush out the requirements in this area.

As far as the authors are aware, there is no combination of tools available that address the primary requirements from this list.

SYSTEM DESIGNER SKETCHPAD (SDS)

Savantage is developing a prototype system physical partitioning environment called the System Designer Sketchpad (SDS) that satisfies the requirements listed above. The architecture of the system is shown in Figure 3. SDS is designed to run under Windows as a client to an analysis and database server that runs under UNIX. A co-application to SDS is an Analysis Cockpit which provides the graphical user interface for controlling tradeoff analysis, reporting, and graphing features.

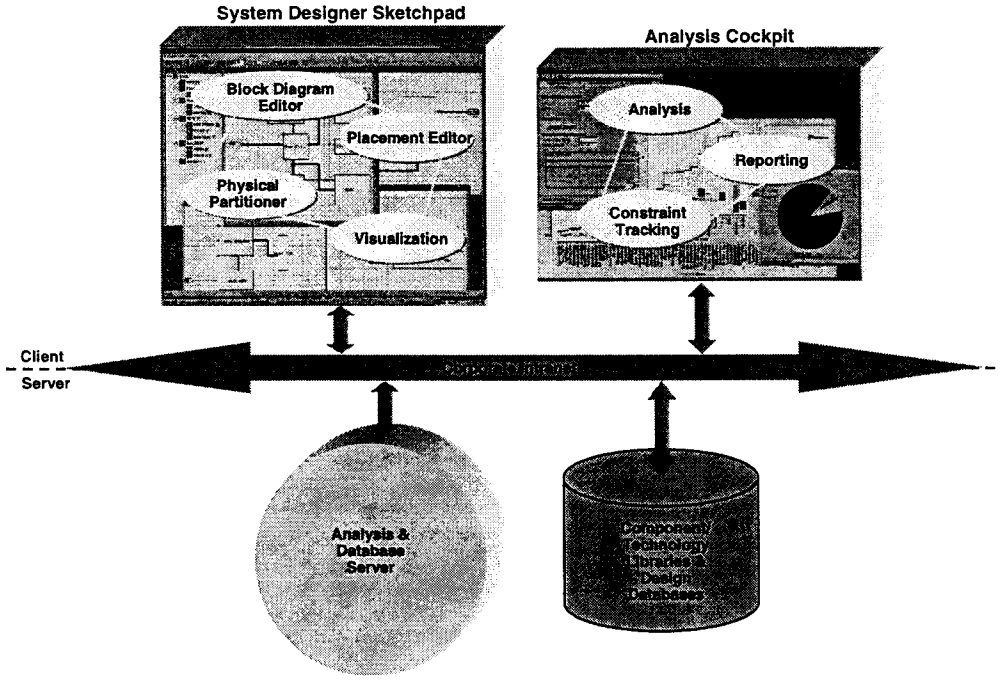

Figure 3: System Designer Sketchpad Architecture

The capability being planned includes:

Hierarchical Block Diagram Capture: The block diagram editor built into SDS is intended to provide a quick and easy way to create and manipulate the logical architecture of a system. The emphasis in the block diagram editor is on ease of use, allowing the designer to quickly sketch the logical organization of the design, preparatory to physical partitioning. The block diagram editor is also used throughout the physical partitioning activity as an interface to elements of the system architecture. Figure 4 shows a screenshot of the block diagram editor in SDS.

- Captures high-level connectivity (the major busses, bundles and critical nets.)
- No predefined symbols or pins required. Interfaces are defined as nets/bundles are connected to blocks.
- Fully hierarchical data and interaction model that supports a combination of top-down and bottom-up design approaches.
- Can read structural VHDL or netlists to import block diagrams.

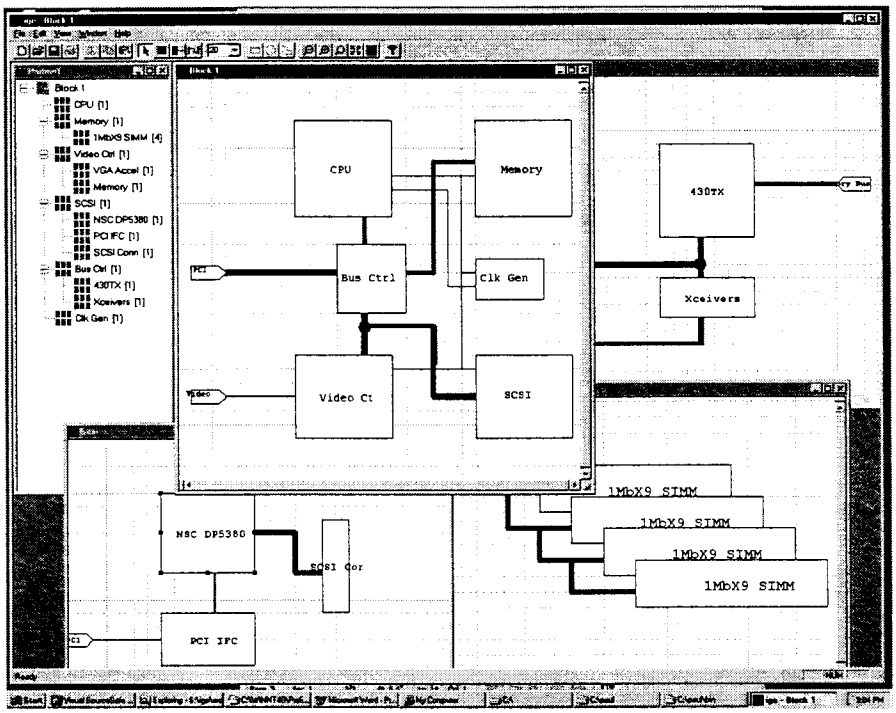

Figure 4: SDS Block Diagram Editor

Physical Partitioning: The SDS Physical Partitioner is intended to provide the tools required to easily move from the functional description of the design to a complete physical architecture. The physical partitioning capability includes the assignment of functional blocks to assemblies, sub-assemblies, modules, and IC's. Physical partitioning can also proceed into sub-assembly placement, component placement, and technology and constraint definition.

- Supports defining assemblies, sub-assemblies, PWBs, regions, modules, and ICs as physical partitions. Each of these partition types may appear at any level in the physical architecture hierarchy. Technology and packaging information for partitions (e.g. layer stackups, design rules, materials definitions, packages, etc.) can be input directly into SDS or read from libraries.
- Supports assignment of functional blocks at any level of the logical hierarchy to physical partitions, and the movement of functionality between physical partitions. Critical metrics, such as partition I/O count and size, can be monitored as functionality is moved around.
- Supports quickly selecting off-the-shelf parts and sub-assemblies for re-use as implementation for functionality in the design. Definition can be black-box (interface information only) or detailed. Physical partitions, technology, and placement can be imported from commercial physical design systems.
- Provides 2.5D placement editor for sub-assembly and component placement. Components can be placed in 3-space, even though the placement editor only provides a 2D view. Editor also allows specification of critical net routing. Connectivity driven placement is supported through rat's nest display. Placement and routing are coarse, intended only to aid in analysis and visualization.

Evaluation: Evaluation of the virtual prototype will not occur directly in SDS, but in the Analysis Cockpit. However, tradeoff analysis requires that changes be made in the design and then re-evaluated. SDS must support quick and easy changes. The Analysis Cockpit and SDS will communicate in order to support the tools' cooperation.

- Perform tradeoff analysis in all of the areas currently supported by SavanSys (see Appendix).
- Provides additional links to 3rd party tools for other analyses not provided by the SavanSys server.

- Provides for quickly making changes in the physical partitioning, implementation or technology selection and evaluating the results of those changes.

Output: Output from SDS has two objectives. The first is to support the physical partitioning and implementation task by creating a virtual prototype that can be examined visually and by electronic analysis. The second is to be able to communicate the final design decisions to downstream detailed design activities as a specification. To meet these objectives, SDS provides the following output capabilities:

- 2D output for visual inspection and communication, and 3D virtual reality output of the system assemblies for visual inspection, most likely using VRML.
- Block Diagram output for presentation (graphical output), and electronically as an annotated block diagram level netlist in structural VHDL and EDIF.
- Design rules, layer stackups, and component placement information export to commercial physical design systems.
- "System Specification" report which contains the bread-crumb trail through the tradeoff analysis in addition to the final conclusions reached.

Figure 5 shows an example of the kinds of output available from the Analysis Cockpit.

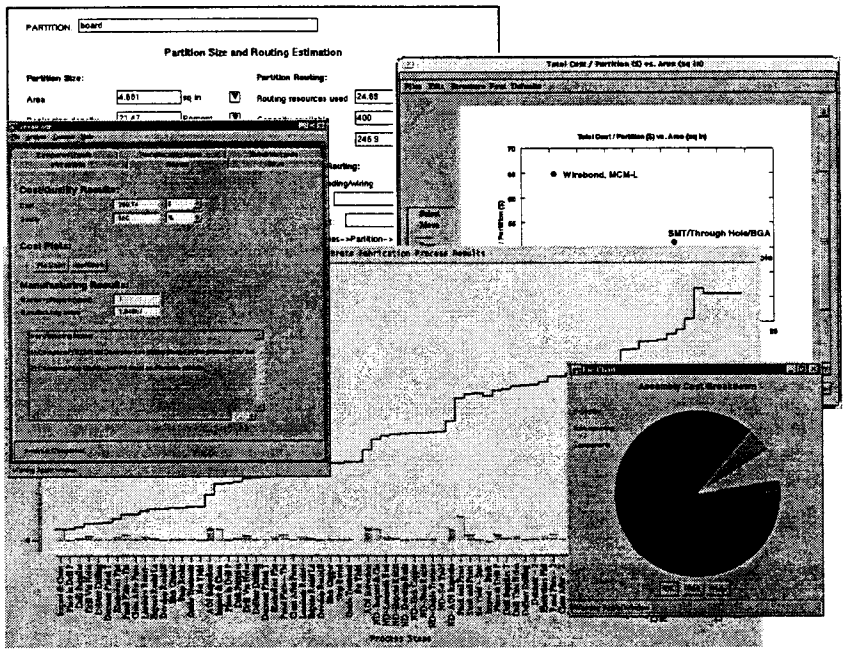

Figure 5: Examples of types of output available from Analysis Cockpit.

SUMMARY

This paper outlined the importance of system physical partitioning as a component of system virtual prototyping that brings analysis and optimization of basic packaging and physical partitioning tradeoffs to the front of the system design process. Combining flexible physical architecture definition capabilities with multi-dimensional analysis to optimize the system characteristics is the central theme of the physical partitioning concept. Frequently the best opportunities to improve products lie in unexplored regions of the physical design space while large amounts of effort are expended in relatively ineffective optimization exercises elsewhere. Tools to enable rigorous physical partitioning do not yet exist and are needed. The requirements for a physical partitioning environment were discussed and the System Designer Sketchpad, a prototype under development at Savantage, was presented as a solution.

APPENDIX—THE SAVANSYS™ SYSTEM PACKAGING TRADEOFF TOOL

SavanSys is a software tool for enhancing the manufacturability and decreasing the design risk associated with the selection of packaging technologies for integrated circuits. The SavanSys software tool performs system packaging tradeoff analysis. SavanSys concurrently computes physical (size, weight, interconnect routing requirements, escape routing), electrical (delays, attenuation, dc drops, effective inductance), thermal (internal and external thermal resistances, air cooling), reliability (MTTF), and cost/yield performance metrics for multichip systems.

Multichip modules (MCMs) and traditional packaging (through-hole and surface mounting) technologies treated by SavanSys include: traditional and fine-line printed circuit boards, low temperature cofired ceramic, and thin-film (chip-first and chip-last). Component

assembly approaches include wirebonding, TAB, flip chip, and single chip packages. Materials are also available for bare die attach, encapsulation, attaching extrusions, and for defining the bonding and substrate technologies.

SavanSys provides the user the ability to compute the cost of assembled electronic systems, including component costs, component preparation (wafer and die level burn-in, bumping), single chip package costs, surface mount and through-hole assembly costs, bare die attach costs (TAB, wirebond, flip chip), tooling costs associated with the processes above, substrate costs, repair and rework costs, and test costs. In addition, learning curves may optionally be defined for any or all steps in the process, and handling costs may be defined for all steps that involve the insertion of components into the process flow.

The SavanSys tradeoff analysis tool is specifically designed to allow the impact of technology, material, and design rule variations on the cost and performance of a board or system of boards. SavanSys enables designers to make optimum physical implementation and physical partitioning choices early in the design process to facilitate successful implementation decisions. SavanSys is integrated into the Mentor Graphics and Cadence physical design frameworks and is compatible with Aspect and DIE format databases.

ACKNOWLEDGMENTS

The authors acknowledge Janet Wedgwood and Linda Madres of Lockheed Martin Advanced Technology Laboratories for their help in defining the physical partitioning activity and the requirements for a physical partitioning tool. The authors also acknowledge Chet Palesko and Steve Spencer at Savantage for their contribution to the development of the SDS prototype.

REFERENCES

Wilson, D. K., 1982. "Topological Aspects of Systems Partitioning," Proc. of Design Policy Conference, Royal College of London, pp. 148-154.

Sandborn, Peter, and A. Parikh, 1995. "Tradeoff Analysis and Partitioning in Multiple Board/MCM Systems", Proceedings of the International Conference on Multichip Modules, pp. 401-406.

Vertal, M. and J. Crowley, 1997. "An Approach to Hardware/Software Comodeling For Rapid Design Exploration", Proceedings of the High-level Electronic System Design Conference 1997, San Jose, CA, October 7-9, 1997.

Sandborn, Peter, 1996. "Optimizing the Design of High-Density Systems", Electronic Packaging & Production, September, 1996, pp. 53-60.

AUTHORS' BIOGRAPHY

Paul Stoaks is a Senior Software Engineer at Savantage, Inc., an electronic design automation (EDA) company that provides solutions and assistance to companies solving difficult electronic system packaging and interconnect problems. With 8 years of experience in the EDA industry, he has worked as a software developer and product engineering manager on IC layout, EDA tool integration, and systems level planning software projects. Paul is currently responsible for the System Designer Sketchpad prototype project at Savantage.

Dr. Peter Sandborn is the Chief Technical Officer and a co-founder of Savantage, Inc. Dr. Sandborn is an expert in electronic packaging design and technology. Dr. Sandborn has done extensive work in CAD tool development for electrical simulation, tradeoff analysis, and cost/manufacturability modeling. At Savantage, Dr. Sandborn manages government programs, web-based software development, and contributes to ongoing CAD tool development. Prior to co-founding Savantage, Inc., Dr. Sandborn was a Sr. Member of Technical Staff at the Microelectronics and Computer Technology Corporation (MCC). Dr. Sandborn has written one book on the design of multichip packaging. Dr. Sandborn holds a doctorate in electrical engineering from the University of Michigan.